

本节内容

# 定点数的 加减运算

关注公众号【研途小时】获取后续课程完整更新！

## 本节总览

### 定点数的加减运算

原码的加减（了解即可）

补码的加减（重要）

溢出判断

注：计算机中，不会用“反码”进行加减运算，因此不作探讨

# 原码的加减运算

原码表示的有符号数

$$\begin{array}{r} 14 \\ -14 \\ + \\ 00001110 \\ 10001110 \\ \hline 10011100 \end{array}$$

加法器直接对原码进行  
加法运算，可能出错

☹

$$\begin{array}{r} 00001110 \\ + 10001110 \\ \hline \end{array}$$

用减法器实现

$$\begin{array}{r} 00001110 \\ - 00001110 \\ \hline 00000000 \end{array}$$

☺

原码的加法运算：

- 正+正 → 绝对值做加法，结果为正
- 负+负 → 绝对值做加法，结果为负
- 正+负 → 绝对值大的减绝对值小的，符号同绝对值大的数
- 负+正 → 绝对值大的减绝对值小的，符号同绝对值大的数

可能会溢出

原码的减法运算，“减数”符号取反，转变为加法：

- 正-负 → 正+正
- 负-正 → 负+负
- 正-正 → 正+负
- 负-负 → 负+正

# 补码的加减运算

设机器字长为8位（含1位符号位）， $A = 15$ ， $B = -24$ ，求 $[A+B]_{\text{补}}$ 和 $[A-B]_{\text{补}}$

	原码	补码
$A = +1111$	$\rightarrow 0,0001111$	$\rightarrow 0,0001111$
$B = -11000$	$\rightarrow 1,0011000$	$\rightarrow 1,1101000$

负数补 $\rightarrow$ 原：①数值位取反+1；②负数补码中，最右边的1及其右边同原码。最右边的1的左边同反码

$$[A+B]_{\text{补}} = [A]_{\text{补}} + [B]_{\text{补}} = 0,0001111 + 1,1101000 = 1,1110111$$

原码：1,0001001 真值-9

$$[A-B]_{\text{补}} = [A]_{\text{补}} + [-B]_{\text{补}} = 0,0001111 + 0,0011000 = 0,0100111 \quad \text{真值}+39$$

$[-B]_{\text{补}}$ ：  $[B]_{\text{补}}$ 连同符号位一起取反加1

$C = 124$ ，求 $[A+C]_{\text{补}}$ 和 $[B-C]_{\text{补}}$

$$[A+C]_{\text{补}} = 0,0001111 + 0,1111100 = 1,0001011$$

真值-117

$$[B-C]_{\text{补}} = 1,1101000 + 1,0000100 = 0,1101100$$

真值+108

溢出



原来如此

对于补码来说，无论加法还是减法，最后都会转变成加法，由加法器实现运算，符号位也参与运算

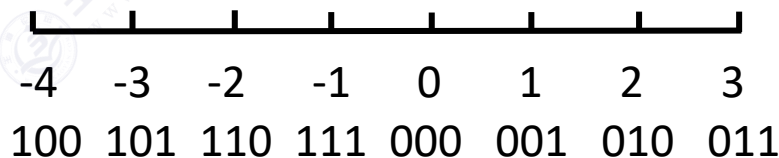
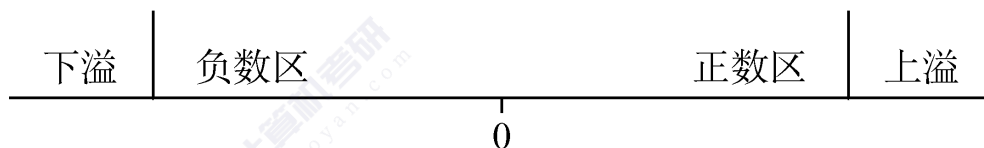
# 溢出判断



设机器字长为8位（含1位符号位）， $A = 15$ ， $B = -24$ ，求 $[A+B]_{\text{补}}$ 和 $[A-B]_{\text{补}}$

$C = 124$ ，求 $[A+C]_{\text{补}}$ 和 $[B-C]_{\text{补}}$

$$\begin{aligned} [A+C]_{\text{补}} &= \boxed{0}0001111 + \boxed{0}1111100 = \boxed{1}0001011 && \text{真值}-117 \\ [B-C]_{\text{补}} &= \boxed{1}1101000 + \boxed{1}0000100 = \boxed{0}1101100 && \text{真值}+108 \end{aligned}$$



只有“正数+正数”才会上溢 —— 正+正=负  
只有“负数+负数”才会下溢 —— 负+负=正

## 溢出判断

设机器字长为8位（含1位符号位）， $A = 15$ ， $B = -24$ ，求 $[A+B]_{\text{补}}$ 和 $[A-B]_{\text{补}}$

$C = 124$ ，求 $[A+C]_{\text{补}}$ 和 $[B-C]_{\text{补}}$

$$[A+C]_{\text{补}} = 00001111 + 01111100 = 10001011$$

真值-117

$$[B-C]_{\text{补}} = 11101000 + 10000100 = 01101100$$

真值+108

### 逻辑表达式

与：如 $ABC$ ，表示A与B与C

仅当A、B、C均为1时， $ABC$ 为1

A、B、C中有一个或多个为0，则 $ABC$ 为0

或：如 $A+B+C$ ，表示A或B或C

仅当A、B、C均为0时， $A+B+C$ 为0

A、B、C中有一个或多个为1，则 $A+B+C$ 为1

非：如 $\bar{A}$ ，表示A非

若A为1，则 $\bar{A}$ 为0

若A为0，则 $\bar{A}$ 为1

方法一：采用一位符号位

设A的符号为 $A_S$ ，B的符号为 $B_S$ ，运算结果的符号为 $S_S$ ，则溢出逻辑表达式为

$$V = A_S B_S \bar{S}_S + \bar{A}_S \bar{B}_S S_S$$

若 $V=0$ ，表示无溢出；

若 $V=1$ ，表示有溢出。

$A_S$ 为1且 $B_S$ 为1且 $S_S$ 为0

或

$A_S$ 为0且 $B_S$ 为0且 $S_S$ 为1



$C = 124$ , 求 $[A+C]_{\text{补}}$ 和 $[B-C]_{\text{补}}$

$$[B-C]_{\text{补}} = 1,1101000 + 1,0000100 = 0,1101100 \quad \text{真值} + 108$$

上溢	0	1
下溢	1	0

处理“不同”的逻辑符号：异或 $\oplus$

若 $V=0$ ，表示无溢出； $V=1$ ，表示有溢出。

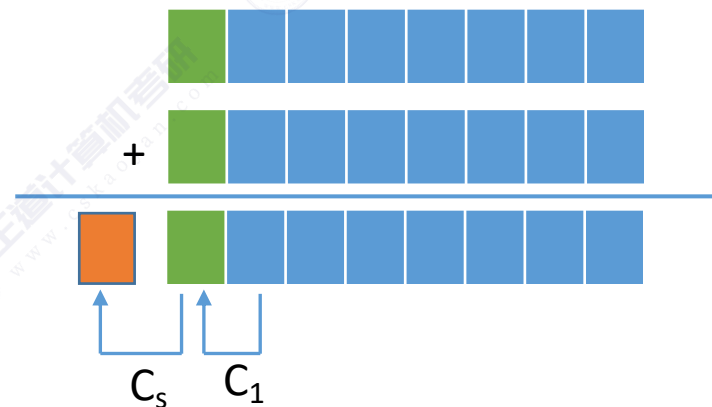
异或逻辑：不同为1，相同为0

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$



## 溢出判断

设机器字长为8位（含1位符号位）， $A = 15$ ， $B = -24$ ，求 $[A+B]_{\text{补}}$ 和 $[A-B]_{\text{补}}$

$C = 124$ ，求 $[A+C]_{\text{补}}$ 和 $[B-C]_{\text{补}}$

$$[A+C]_{\text{补}} = 0,0001111 + 0,1111100 = 1,0001011 \quad \text{真值}-117$$

$$[B-C]_{\text{补}} = 1,1101000 + 1,0000100 = 0,1101100 \quad \text{真值}+108$$

方法三：采用双符号位

正数符号为00，负数符号为11

$$[A+C]_{\text{补}} = 00,0001111 + 00,1111100 = 01,0001011 \quad \text{上溢}$$

$$[B-C]_{\text{补}} = 11,1101000 + 11,0000100 = 10,1101100 \quad \text{下溢}$$

记两个符号位为 $S_{S1}S_{S2}$ ，则 $V = S_{S1} \oplus S_{S2}$

若 $V=0$ ，表示无溢出；若 $V=1$ ，表示有溢出。

$$[A+B]_{\text{补}} = 00,0001111 + 11,1101000 = 11,1110111$$

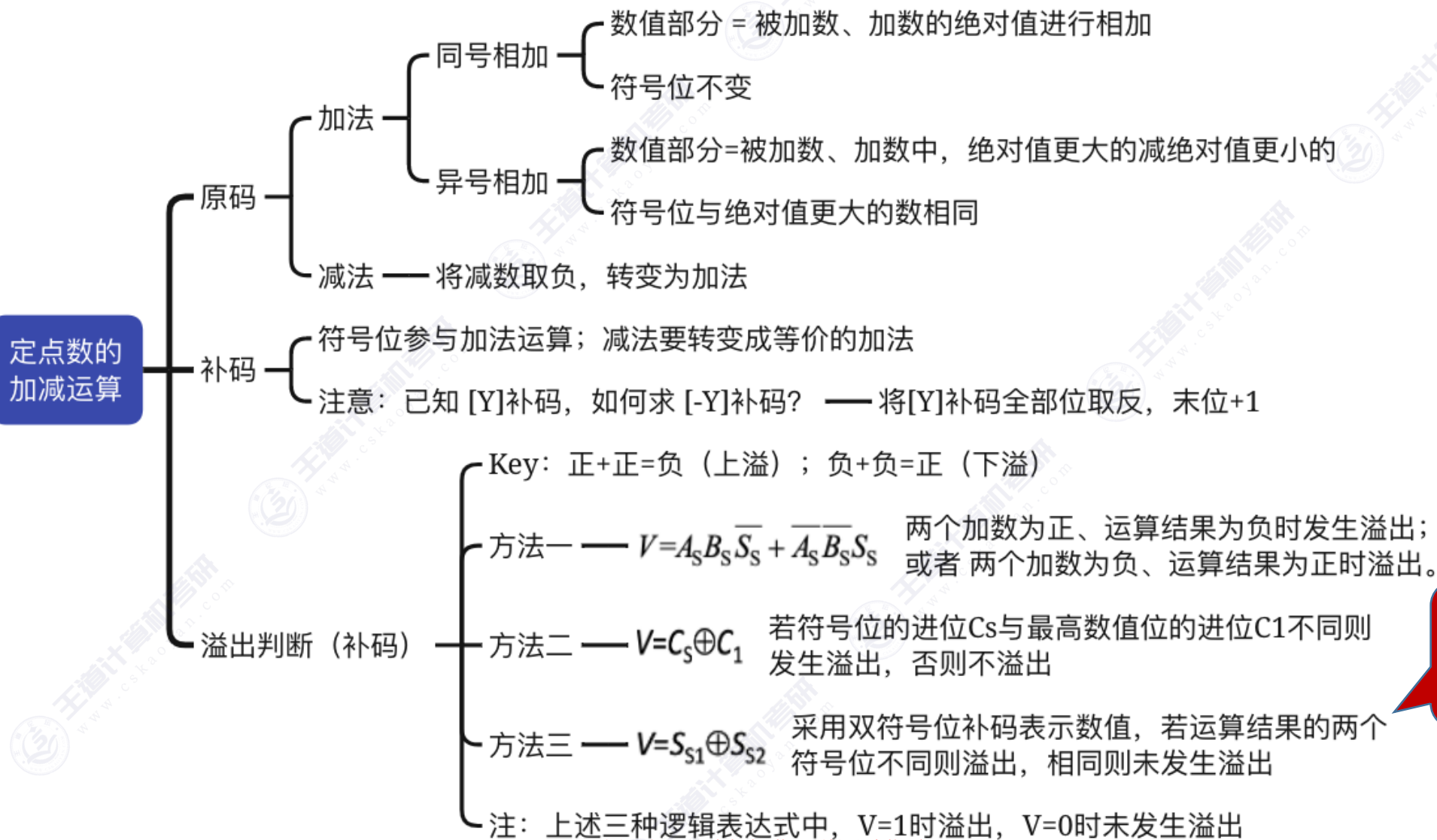
$$[A-B]_{\text{补}} = 00,0001111 + 00,0011000 = 00,0100111$$

实际存储时只存储1个符号位，运算时会复制一个符号位

双符号位补码又称：模4补码  
单符号位补码又称：模2补码



# 知识点回顾



模4补码、  
模2补码的  
概念